Cite No. 2.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
18 July 2002 (18.07.2002)

PCT

(10) International Publication Number
WO 02/056297 A1

(51) International Patent Classification[7]: G10L 19/02, 19/00

(21) International Application Number: PCT/IB01/01371

(22) International Filing Date: 31 July 2001 (31.07.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/261,358 11 January 2001 (11.01.2001) US

(71) Applicant *(for all designated States except US)*: **SASKEN COMMUNICATION TECHNOLOGIES LIMITED** [IN/IN]; HAL 2nd Stage, 5008 12th B Main, Indiranagar, Bangalore 560 008, Karnataka (IN).

(71) Applicants and
(72) Inventors: **CHAKRAVARTHY, K., P., P., Kalyan** [IN/IN]; HAL III Stage, 361, 6th Cross, 4th Main, Bangalore 560 075, Karnataka (IN). **RUTHRAMOORTHY, Navaneetha, K.** [IN/US]; Apartment 7122, 1400 Worcester Road, Framingham, MA 01702 (US). **PATWARDHAN, Pushkar, P.** [IN/IN]; Shreerang Society, CD-65 C11, Thanewest 400 601 (IN). **MONDAL, Bishwarup** [IN/IN]; 79, Kalitala Road, Purbachal, Kalikapur, Kolkata 700078 (IN).

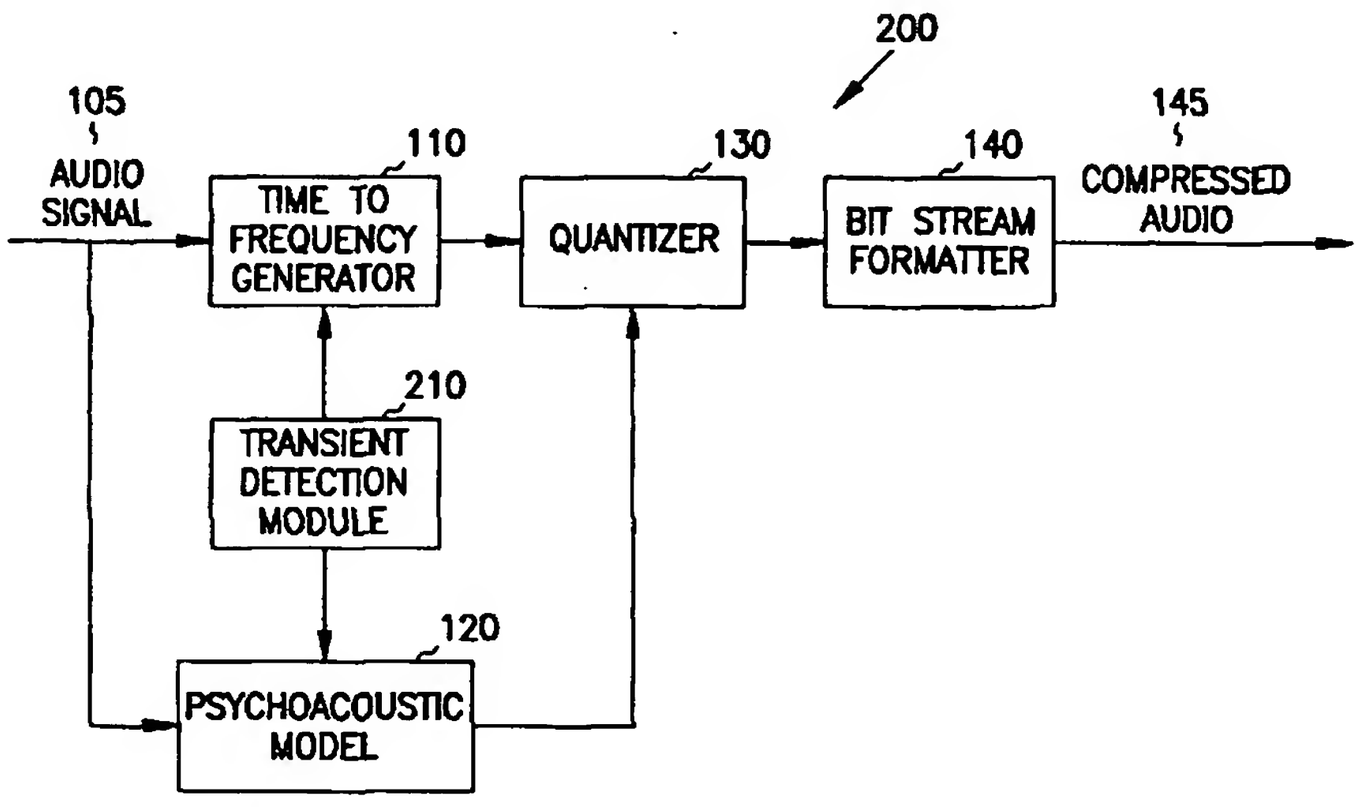
(74) Agent: **VIKSNINS, Ann, S.**; Schwegman, Lundberg, Woessner & Kluth, P.O. Box 2938, Minneapolis, MN 55402 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European

[Continued on next page]

(54) Title: COMPUTATIONALLY EFFICIENT AUDIO CODER



(57) Abstract: The present invention provides a computationally efficient technique for compression encoding of an audio signal, and further provides a technique to enhance the sound quality of the encoded audio signal. This is accomplished by including more accurate attack detection and a computationally efficient quantization technique. The improved audio coder converts the input audio signal to a digital audio signal. The audio coder then divides the digital audio signal into larger frames having a long-block frame length and partitions each of the frames into multiple short-blocks. The audio coder then computes short-block audio signal characteristics for each of the partitioned short-blocks based on changes in the input audio signal. The audio coder further compares the computed short-block characteristics to a set of threshold values to detect presence of an attack in each of the short-blocks and changes the long-block frame length of one or more short-blocks upon detecting the attack in the respective one or more short-blocks.

WO 02/056297 A1

patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

ADAPTIVE-BL0CK-LENGTH AUDIO CODER

Field of the Invention

This invention relates generally to processing of information signals and more particularly pertains to techniques for encoding audio signals inclusive of voice and music using a perceptual audio coder.

Background

A Perceptual audio coder is an apparatus that takes series of audio samples as input and compresses them to save disk space or bandwidth. The Perceptual audio coder uses properties of the human ear to achieve the compression of the audio signals.

The technique of compressing audio signals involves recording an audio signal through a microphone and then converting the recorded analog audio signal to a digital audio signal using an A/D converter. The digital audio signal is nothing but a series of numbers. The audio coder transforms the digital audio signal into large frames of fixed-length. Generally, the fixed length of each large frame is around 1024 samples. The analog signal is sampled at a specific rate (called the sampling frequency) and this results in a series of audio samples. Typically a frame of samples is a series of numbers. The audio coder can only process one frame at a time. This means that the audio coder can process only 1024 samples at a time. Then the audio coder transforms the received fixed-length frames (1024 samples) into a corresponding frequency domain. The transformation to a frequency domain is accomplished by using an algorithm, and the output of this algorithm is another set of 1024 samples representing a spectrum of the input. In the spectrum of samples, each sample corresponds to a frequency. Then the audio coder computes masking thresholds from the spectrum of samples. Masking thresholds are nothing but another set of numbers, which are useful in compressing the audio signal. The following illustrates the computing of masking thresholds.

The audio coder computes an energy spectrum by squaring the spectrum of the 1024 samples. Then the samples are further divided into series of bands.

For example, the first 10 samples can be one band and the next 10 samples can be another subsequent band and so on. Note that the number of samples (width) in each band varies. The width of the bands is designed to best suit the properties of the human ear for listening to frequencies of sound. Then the computed energy spectrum is added to each of the bands separately to produce a grouped energy spectrum.

The audio coder applies a spreading function to the grouped energy spectrum to obtain an excitation pattern. This operation involves simulating and applying the effects of sounds in one critical band to a subsequent (neighboring) critical band. Generally this step involves convolution with a spreading function, which results in another set of fixed numbers.

Then, based on the tonal or noise-like nature of the spectrum in each critical band, a certain amount of frequency-dependent attenuation is applied to obtain initial masking threshold values. Then, by using an absolute threshold of hearing, the final masked thresholds are obtained. Absolute threshold of hearing is a set of amplitude values below which the human ear will not be able to hear.

Then the audio coder combines the initial masking threshold values with the absolute threshold values to obtain the final masked threshold values. Masked threshold value means a sound value below which a sound is not audible to the human ear (i.e., an estimate of maximum allowable noise that can be introduced during quantization).

Using the masked threshold values, the audio coder computes perceptual entropy (PE) of a current frame. The perceptual entropy is a measure of the minimum number of bits required to code a current frame of audio samples. In other words, the PE indicates how much the current frame of audio samples can be compressed. Various types of algorithms are currently used to compute the PE.

The audio coder receives the grouped energy spectrum, the computed masking threshold values, and the PE and quantizes (compresses) the audio signals. The audio coder has only a restricted number of bits allocated for each frame depending on a bit rate. It distributes these bits across the spectrum based on the masking threshold values. If the masking threshold value is high, then the audio signal is not important and is hence represented using a smaller number of

bits. Similarly, if masking threshold is low, the audio signal is important and hence represented using a higher number of bits. Also, the audio coder checks to ensure that the allocated number of bits for the audio signals is not exceeded. The audio coder generally applies a two-loop strategy to allocate and monitor the number of bits to the spectrum. The loops are generally nested and are called Rate Control and Distortion Control Loops. The Rate Control Loop controls the distribution of the bits not to exceed the allocated number of bits, and the Distortion control loop does the distribution of the bits to the received spectrum. Quantization is a major part of the perceptual audio coder. The performance of the audio coder can be significantly improved by reducing the number of calculations performed in the control loops. The current quantization algorithms are very computation intensive and hence result in a slower operation.

Earlier we have seen that the audio coder receives one frame of samples (1024 samples in length) as input and converts the frame of samples into a spectrum and then quantizes using masking thresholds. Sometimes the input audio signal may vary quickly (when the properties of a signal change abruptly). For example, if there is a sudden heavy beat in the audio signal, and if the audio coder receives a frame of 1024 samples in length (including the heavy beat) due to inadequate temporal masking in a signal including abrupt changes, a problem called pre-echo can occur. This is because the sound signal contains error after quantization, and this error can result in an audible noise before the onset of the heavy beat, hence called the pre-echo. Heavy beats are also called 'attacks.' A signal is said to have an attack if it exhibits a significant amount of non-stationarity within the duration of a frame under analysis. For example, sudden increase in amplitudes of a time signal within a typical duration of analysis is an attack. To avoid this problem the audio signal is coded with frames having smaller frame lengths instead of the long 1024 samples. To keep continuity in the number of samples given as input usually 8 smaller blocks of 128 samples are coded (8x128 samples = 1024 samples). This will restrict the heavy beat to one set of 128 samples among 8 smaller blocks, and hence the noise introduced will not spread to the neighboring smaller blocks as pre-echo. But the disadvantage of coding in 8 smaller blocks of 128 samples is that they require more bits to code than required by the larger blocks of 1024 samples in length.

So the compression efficiency of the audio coder is significantly reduced. To improve the compression efficiency, the heavy beats have to be detected accurately so that the smaller blocks can be applied only around the heavy beats. It is important that the heavy beats be accurately detected, or else pre-echo can occur. Also, a false detection of heavy beats can result in significantly reduced compression efficiency. Current methods to detect the heavy beats use the PE. Calculating the PE is computationally very intensive and also not very accurate.

Also, we have seen earlier that the blocks that have attacks should be coded as smaller blocks having 128 samples and others as larger blocks having 1024 samples. The smaller frame lengths of 128 samples are called 'short-blocks', and the 1024 samples frame length are called 'long-blocks.' We have also seen that the short-blocks require more bits to code than the long-blocks. Also for each large frame there is a fixed number of bits allocated. If we can intelligently save some bits while coding a long-block and use the saved bits in a short-block, the compression efficiency of the audio coder can be significantly increased. For storing the bits, a 'Bit Reservoir mechanism' is needed. Since long-blocks do not need a large number of bits, the unused bits from the long-blocks can be saved in the bit reservoir and used later for a short-block. Currently there are no efficient techniques to save and allocate bits between long and short-blocks to improve the compression efficiency of the audio coder.

The audio signal can be of two types (i) single channel or mono-signal and (ii) multi-channel or stereo signal to produce spatial effects. The stereo signal is a multi-channel signal comprised of two channels, namely left and right channels. Generally the audio signals in the two channels have a large correlation between them. By using this correlation the stereo channels can be coded more efficiently. Instead of directly coding the stereo channels, if their sum and difference signals are coded and transmitted where the correlation is high, a better quality of sound is achieved at a same bit rate. When the audio signal is a stereo signal, the audio coder can operate in two modes (a) normal mode and (b) M-S mode. The M-S mode means encoding the sum and difference of the left and right channels of the stereo. Currently the decision to switch between the normal and M-S modes is based on the PE. As explained before, computing PE is very computation intensive and inconsistent.

Therefore, there is a need in the art for a computationally efficient quantization technique. Also, there is a need in the art for an improved attack detection technique that is computationally less intensive and more accurate, to improve the compression efficiency of the audio coder. In addition, there is a need in the art for a technique to allocate the bits between the long and short-blocks to improve the computation efficiency of the audio coder. Furthermore, there is also a need in the art for a technique that is computationally efficient and more accurate in switching between the normal and the M-S modes when the audio signal is a stereo signal.

Summary of the Invention

The present invention provides an improved technique for detecting an attack in an input audio signal to reduce pre-echo artifacts caused by attacks during compression encoding of the input audio signal. This is accomplished by providing a computationally efficient and more accurate attack detection technique. The improved audio coder converts the input audio signal to a digital audio signal. The audio coder then divides the digital audio signal into larger frames having a long-block frame length and partitions each of the frames into multiple short-blocks. The audio coder then computes short-block audio signal characteristics for each of the partitioned short-blocks based on changes in the input audio signal. The audio coder further compares the computed short-block characteristics to a set of threshold values to detect presence of an attack in each of the short-blocks and changes the long-block frame length of one or more short-blocks upon detecting the attack in the respective one or more short-blocks.

Further, the improved audio coder increases compression efficiency by efficiently allocating bits between long and short-blocks. The audio coder that is computationally efficient and more accurate in switching between the normal and M-S modes when the audio signal is a stereo signal. In addition, the present invention also describes a technique for reducing the computational complexity of quantization.

Brief Description of the Drawings

Figure 1 is block diagram of a prior-art perceptual audio coder.

Figure 2 is a block diagram of a perceptual audio coder according to the teaching of the present invention.

Figure 3 is a block diagram of one example embodiment of computing inter-block differences.

Figure 4 is a block diagram of one embodiment of major components of the Quantizer shown in Figure 2 and their interconnections.

Figure 5 is a flowchart illustrating the overall operation of the embodiment shown in Figure 2.

Figure 6 is a flowchart illustrating the operation of the Bit Allocator shown in Figure 4.

Figure 7 is a flowchart illustrating the operation of the Quantizer shown in Figures 1 and 2 according to the teachings of the present invention.

Figure 8 is a flowchart illustrating the overall operation of the embodiment shown in Figure 2 when compression encoding a stereo audio signal according to the teachings of the present invention.

Figure 9 shows an example of a suitable computing system environment for implementing embodiments of the present invention, such as those shown in Figures 1-8.

Detailed Description

The present invention provides an improved audio coder by increasing the efficiency of the audio coder during compression of an input audio signal. This is accomplished by providing computationally efficient and more accurate attack detection and quantization technique. Also, compression efficiency is improved by providing a technique to allocate bits between long and short-blocks. In addition, the present invention provides an audio coder that is computationally efficient and more accurate in switching between the normal and M-S modes when the audio signal is a stereo signal. The words 'encode' and 'code' are used interchangeably throughout this document to represent the same audio compression scheme. Also the words 'encoder' and 'coder' are used

interchangeably throughout this document to represent the same audio compression system.

Figure 1 shows a prior-art perceptual audio coder 100 including major components and their interconnections. Shown in Figure 1 are Time frequency generator 110, Psychoacoustic model 120, Quantizer 130, and BitStream Formatter 140. The technique of compressing audio signals involves recording an audio signal through a microphone and then converting the recorded analog audio signal to a digital audio signal using an A/D converter. The digital audio signal is nothing but a series of numbers.

The Time frequency generator 110 receives the series of numbers in large frames (blocks) of fixed-length 105. Generally, the fixed length of each frame is around 1024 samples (series of numbers). Time frequency generator 110 can only process one frame at a time. This means that the audio coder 100 can process only 1024 samples at a time. The Time frequency generator 110 then transforms the received fixed-length frames (1024 samples) into corresponding frequency domains. The transformation to the frequency domain is accomplished by using an algorithm, and the output of this algorithm is another set of 1024 samples called a spectrum of the input. In the spectrum, each sample corresponds to a frequency. Then the Time frequency generator 110 computes masking thresholds from the spectrum. Masking thresholds are nothing but another set of numbers that are useful in compressing the audio signal. The following illustrates one example embodiment of computing masking thresholds.

The Time frequency generator 110 computes an energy spectrum by squaring the spectrum of 1024 samples. Then the samples are further divided into series of bands. For example, the first 10 samples can be one band and the next 10 samples can be another subsequent band and so on. Note that the number of samples (width) in each band varies. The width of the bands is designed to best suit the properties of the human ear for listening to frequencies of sound. Then the computed energy spectrum is added to each of the bands separately to produce a grouped energy spectrum.

The Time frequency generator 110 then applies a spreading function to the grouped energy spectrum to obtain an excitation pattern. This operation involves simulating and applying the effects of sounds in one critical band to a

subsequent (neighboring) critical band. Generally this step involves using a convolution algorithm between the spreading function and the energy spectrum.

Based on the tonal or noise-like nature of the spectrum in each critical band, a certain amount of frequency dependent attenuation is applied to obtain initial masking threshold values. Using an absolute threshold of hearing, the final masked thresholds are obtained. Absolute threshold of hearing is a set of amplitude values below which the human ear will not be able to hear.

The Psychoacoustic model 120 combines the initial masking threshold values with the absolute threshold values to obtain the final masked threshold values. Masked threshold value means a sound value below which quantization noise is not audible to the human ear (it is an estimate of the maximum allowable noise that can be introduced during quantization).

Using the masked threshold values, the Psychoacoustic model 120 computes perceptual entropy (PE). The perceptual entropy is a measure of the minimum number of bits required to code a current frame of audio samples. In other words, the PE indicates how much the current frame of audio samples can be compressed. Various types of algorithms are currently used to compute the PE.

The Quantizer 130 then receives the spectrum, the computed masking threshold values, and the PE, and compresses the audio signals. The Quantizer 130 has only a specific number of bits allocated for each frame. It distributes these bits across the spectrum based on the masking threshold values. If the masking threshold value is high, then the audio signal is not important and hence can be represented using a smaller number of bits and similarly, if the masking threshold is low, the audio signal is important and hence can only be represented using a higher number of bits. Also, the Quantizer 130 checks to make sure that the allocated number of bits for the audio signals is not exceeded. The Quantizer 130 generally applies a two-loop strategy to allocate and monitor the number of bits to the received spectrum. The loops are generally nested and are called Rate control and Distortion control loops. The Rate Control loop controls the global gain so that the number of bits used to code the spectrum does not exceed the allocated number of bits, and the Distortion control loop does the distribution of the bits to the received spectrum. Quantization is a major part of the perceptual

audio coder 100. The performance of the Quatizer 130 can be significantly improved by reducing the number of calculations performed in the control loops. The current quantization algorithms used in the Quantizer 130 are very computation intensive and hence result in slower operation.

BitStream formatter 140 receives the compressed audio signal (coded bits) from the Quatizer 130 and converts it into a desired format/syntax (specified coding standard) such as ISO MPEG-2 AAC.

Figure 2 is a block diagram of one embodiment of a perceptual audio coder 200 according to the teachings of the present invention. In addition to what is shown in Figure 1, in this embodiment the perceptual audio coder 200 includes a transient detection module 210. The transient detection module is coupled to receive the input audio signal. Also, the transient detection module 210 is coupled to provide an input to the time frequency generator 110 and psychoacoustic model 120.

In operation, the transient detection module 210 receives the input audio signal 105 as a series of numbers in frames of fixed-length and partitions each of the frames into multiple short-blocks. In some embodiments, the fixed length is a long-block frame length of 1024 samples of digital audio signal. The digital audio signal comprises series of numbers. The long-block is used when there is no attack in the input audio signal. In some embodiments, the short-blocks have a frame length in the range of about 100 to 300 samples of digital audio signal.

The transient detection module 210 computes short-block audio signal characteristics for each of the partitioned short-blocks. In some embodiments, computing the short-block audio signal characteristics includes computing inter-block differences (*xdiff(m)* for an *mth* short-block) and inter-block ratios, and further determining maximum inter-block difference and ratio, respectively. In some embodiments, computing the inter-block differences includes summing a square of the differences between samples in adjacent short-blocks. Further, in some embodiments, the inter-block ratios are computed to better isolate (detect) the attacks. In this embodiment, the inter-block ratios are computed by dividing the adjacent computed inter-block differences as follows:

```
r[0] = xdiff[0]/pxdif
```

$$r[1] = xdiff[1]/xdiff[0]$$
$$r[2] = xdiff[2]/xdiff[1]$$
$$r[3] = xdiff[3]/xdiff[2]$$
$$r[4] = xdiff[4]/xdiff[3]$$

where '*pxdif* is $xdiff_p[4]$ (which is *xdiff[4]* of the previous frame)

The transient detection module 210 compares the computed short-block characteristics with a set of threshold values to detect the presence of an attack in each of the short-blocks. Then the transient detection module 210 changes the long-block frame length of the frame including the attack based on the outcome of the comparison, and inputs the changed frame length to the time frequency generator 110 to reduce the effect of the pre-echo caused by the attack. In some embodiments, the time frequency generator uses short-blocks to restrict the attack to a smaller frame so that the attack does not spread to adjacent smaller frame lengths to reduce the pre-echo artifact caused by the attack. In this embodiment, the smaller frames have a frame length in the range of about 100 to 200 samples of digital audio signal.

Figure 3 illustrates an overview of one embodiment of computing inter-block differences to detect the presence of an attack in an input audio signal according to the teachings of the present invention. As explained earlier with reference to Figures 1 and 2, the input audio signal 305 is divided into large frames by a signal splitter 330 and processed by the perceptual audio coder 200 into frames. Each of the frames has a long-block frame length of 1024 samples of digital audio signal. The transient detection module 210 detects the presence of an attack by using two adjacent incoming frames at a time. In the example embodiment shown in Figure 3 the transient detection module 210 receives two adjacent current and previous frames 310 and 320, respectively. Also shown are the partitioned short-blocks 315 and 325 corresponding to the frames 310 and 320, respectively. In the embodiment shown in Figure 3, each of the short-blocks 315 and 325 corresponding to the frames 310 and 320, respectively, have frame lengths of 256 samples. The last five short-blocks (the four short-blocks 315 from the frame 310 and one adjacent short-block 325 from the frame 320) are

used in detecting the presence of an attack in the adjacent frame 320 before transformation to frequency domain by the Time frequency generator 110.

The following computational sequence is used in detecting the presence of an attack in the adjacent frame 320:

The inter block differences xdiff(m) 340 in the time domain are computed using the following algorithm:

$$xdiff(m) = \frac{4}{N}\sum_{j=0}^{N/4-1}[s(j,m) - s(j,m-1)]^2$$

where *s(j.m)* is the *j'th* time domain sample of the *m'th* short-block and *s(j,m-1)* corresponds to time domain samples of the last short-block of the adjacent frame 320. The Diff blocks 350 shown in Figure 3 compute the difference between two adjacent short-blocks 315 and 325. The $(\,)^2$ blocks 360 in Figure 3 compute the square of the respective computed differences. The Σ blocks 370 compute the sum, and finally the *xdiff(m)* is computed as indicated in the above algorithm.

In some embodiments, the short-block frame lengths are tuned to the application in use. In these embodiments, distance between the large frames is computed to determine an optimum size for the short-block frame lengths. The following algorithm is used to compute the distance between the large frames:

$$xdiff(m) = d(\hat{S}_m, \hat{S}_{m-1})$$

where $\hat{S}_m$ *and* $\hat{S}_{m-1}$ 380 are the signal sub-vectors for the m^{th} and $(m-1)^{th}$ short-blocks, and d (.) is a function that returns a distance measure between the two vectors.

Figure 4 illustrates one embodiment of the major components of the Quantizer 130 and their interconnections as shown in Figure 2 used in a bit allocation strategy according to the teachings of the present invention. Shown in Figure 4 are Bit Allocator 410, Bit Reservoir 420, and Memory 425. The

technique of bit allocation strategy according to the teachings of the present invention includes efficient distribution of bits to different portions of the audio signal. Bits required to code the current frame can be estimated from the perceptual entropy of that frame. Extensive experimentation suggests that the number of bits required to encode is considerably less for a larger frame length than for a smaller frame length. Also, it has been found that the larger frames generally require less than the average number of bits to encode large frames. The amount of reduction below the average number of bits is a function of bit rate. Using this technique also results in large savings of bits during stationary portions of the audio signal. The technique of bit allocation strategy according to the teachings of the present invention is explained in detail in the following section.

The Quantizer 130 receives the large and small frames including the samples of digital audio signal from the time frequency generator 110. Further, the Quantizer 130 receives the computed perceptual entropy from the psychoacoustic model 120 shown in Figure 2. The Bit Allocator 410 computes an average number of bits that can be allocated to each of the received large frames. In some embodiments, the Bit Allocator 410 determines the average number of bits by using the long-block frame length and sampling frequency of the input audio signal. Further, the Bit Allocator 410 computes a bit rate and a reduction factor based on the computed bit rate, and the received perceptual entropy. In addition, the Bit Allocator 410 computes a reduced average number of bits that can be allocated for each of the large frames using the computed reduction factor. Further, the Bit Allocator 410 computes remaining bits by subtracting the computed average number of bits using the computed reduced average number of bits. The Bit Allocator 410 includes a Bit Reservoir 420 to receive the remaining bits. The Bit Allocator 410 allocates a reduced average number of bits to the current frame and stores the remaining bits in the Bit Reservoir 420 when the current frame is a large frame. Further, the Bit Allocator allocates the reduced number of bits along with the stored bits from the Bit Reservoir 420 when the current frame is a small frame to improve the bit allocation between the large and small frames, to enhance sound quality of the compressed audio signal. The Bit Allocator 410 repeats the above process of bit

allocation to a next adjacent frame. In some embodiments, the allocation of bits to a small frame is based on number of bits available in the Bit Reservoir 420, bit rate, and a scaling applied to the denominator, which actually distributes the bits across continuous sequence of frames that use finer time resolution. At the same time, the Bit Allocator 410 makes sure that the Bit Reservoir 420 is not depleted too much.

Figure 4 also illustrates one embodiment of major components and their interconnections in the Quantizer 130 shown in Figure 2 used in reducing computational complexity in the Quantizer 130 according to the teachings of the present invention. Also shown in Figure 4 are Rate Control Loop 430 (also generally referred to as "Inner Iteration Loop"), Comparator 427, and Distortion Control Loop 440 (also generally referred to as "Outer Iteration Loop").

The Rate Control Loop 430 computes global gain, which is commonly referred to as "common scalefac" for a given set of spectral values with a pre-determined value for the maximum number of bits available for encoding the frame (referred to as "available bits"). The Rate Control Loop arrives at a unique solution for the common scalefac value for a given set of spectral data for a fixed value of available bits, so any other variation of the Rate Control Loop must necessarily arrive at the same solution. Efficiency of the Rate Control Loop is increased by reducing the number of iterations required to compute the common scalefac value. The technique of reducing the number of iterations required to compute the common scalefac value according to the teachings of the present invention is discussed in detail in the following section.

The Quantizer 130 stores a start common scalefac value of a previous adjacent frame to use in quantization of a current frame. The Rate Control Loop 430 computes the common scalefac value for the current frame using the stored start common scalefac value as a starting value during computation of iterations by the Rate Control Loop 430 to reduce the number of iterations required to compute the common scalefac value of the current frame. Further, the Rate control Loop 430 computes counted bits using the common scalefac value of the current frame. The comparator 427 coupled to the Rate control Loop compares the computed count bits with available bits. The Rate Control Loop changes the computed common scalefac value based on the outcome of the comparison. In

some embodiments, the count bits comprises bits required to encode a given set of spectral values for the current frame.

The Distortion Control Loop 440 is coupled to the Rate Control Loop 430 to distribute the bits among the samples in the spectrum based on the masking thresholds received from the psychoacoustic model. Also, the Distortion Control Loop 440 tries to allocate bits in such a way that quantization noise is below the masking thresholds. The Distortion Control Loop 440 also sets the starting value of start common scalefac to be used in the Rate Control Loop 430.

Figure 5 illustrates one example embodiment of a process 500 of detecting an attack in an input audio signal to reduce a pre-echo artifact caused by the attack during a compression encoding of the input audio signal. The process 500 begins with step 510 by receiving an input audio signal and converting the received input audio signal into a digital audio signal. In some embodiments, the attack comprises a sudden increase in signal amplitude.

Step 520 includes dividing the converted digital audio signal into large frames having a long-block frame length. In some embodiments, the long-block frame length comprises 1024 samples of digital audio signal. In this embodiment, the samples of digital audio signal comprise series of numbers. In this embodiment, the long-block frame length comprises a frame length used when there is no attack in the input audio signal.

Step 530 includes partitioning each of the large frames into multiple short-blocks. In some embodiments, partitioning large frames into short-blocks includes partitioning short-blocks having short-block frame lengths in the range of about 100 to 300 samples.

Step 540 includes computing short-block characteristics for each of the partitioned short-blocks based on changes in the input audio signal. In some embodiments, the computing of the short-block characteristics includes computing inter-block differences and determining a maximum inter-block difference from the computed inter block differences. In some embodiments, the computing of short-block characteristics further includes computing inter-block ratios and determining a maximum inter-block ratio from the computed inter-block ratios. In this embodiment, the computing of inter-block differences

includes summing a square of the differences between samples in adjacent short-blocks. Also in this embodiment the computing of the inter-block ratios includes dividing the adjacent computed inter-block differences. The process of computing the short-block characteristics is discussed in more detail with reference to Figure 3.

Step 550 includes comparing the computed short-block characteristics to a set of threshold values to detect a presence of the attack in each of the short-blocks. Step 560 includes changing the long-block frame length of one or more large frames based on the outcome of the comparison to reduce the pre-echo artifact caused by the attack. In some embodiments, the changing of the long-block frame length means changing to include multiple smaller frames to restrict the attack to one or more smaller frames so that the pre-echo artifact caused by the attack does not spread to the adjacent larger frames. In some embodiments, the smaller frame lengths include about 100 to 200 samples of digital audio signal.

Figure 6 illustrates one example embodiment of an operation 600 of an efficient strategy for bit allocation to the large and small frames by the Bit Allocator shown in Fig. 4 according to the present invention. The operation 600 begins with step 610 by computing an average number of bits that can be allocated for each of the large frames. In some embodiments, the average number of bits is computed by determining the long-block frame length, the sampling frequency of the input audio signal, and the bit rate of the coding the input audio signal.

Step 620 includes computing a perceptual entropy for the current frame of audio samples using the masking thresholds computed as described in detail with reference to Figure 1. Step 630 includes computing a bit rate using a sampling frequency and the current frame length. Step 640 includes computing a reduction factor based on the computed bit rate and the perceptual entropy. Step 650 includes computing a reduced average number of bits that can be allocated to each of the large frames using the computed reduction factor. Step 660 includes computing remaining bits by subtracting the computed average number of bits with the computed reduced average number of bits. Step 670 includes allocating bits based on the large or small frame. In some embodiments, if the

current frame to be coded is large, then a reduced number of bits are allocated to the current frame and the remaining bits are stored in a Bit Reservoir, and if the current frame to be coded is small, then the reduced number of bits are allocated along with the stored bits from the Bit Reservoir. In some embodiments, the above-described operation 600 repeats itself for a next frame adjacent to the current frame.

The following example further illustrates the operation of the above-described operation 600 of the bit allocation strategy:

For example, if a given mono (single) audio signal at a bit rate of 64kbps is sampled at a sampling frequency of 44100 Hz (meaning there are 44100 samples per second which needs to be encoded at a bit rate of 64000 bits per second) and the long-block frame length is 1024 samples, the average number of bits are computed as follows:

$$\text{Average number of bits} = \frac{64000 * 1024}{44100} = 1486.08 \sim 1486$$

Therefore each frame is coded using 1486 bits. Each of the frames does not require the same number of bits. Also each of the frames does not require all of the bits. Assuming the first frame to be coded requires 1400 bits, the remaining unused 86 bits are stored in the Bit Reservoir and can be used in succeeding frames. For the next adjacent frame we will have a total of 1572 bits (1486 bits + 86 bits in the Bit Reservoir) available for coding. For example, if the next adjacent frame is a short frame more bits can be allocated for coding.

In some embodiments, less than the average number of bits are used for encoding the large frames (using a reduction factor) and the remaining bits are stored in the Bit Reservoir. For example, in the above case only 1300 bits are allocated for each of the large frames. Then the remaining 186 bits (reduction factor) are stored in the Bit Reservoir.

Generally the Bit Reservoir cannot be used to store a large number of remaining bits. Therefore, a maximum limit is set for the number of bits that can be stored in the Bit Reservoir, and anytime the number of bits exceeds the

maximum limit, the excess bits are allocated to the next frame. In the above example, if the bit reservoir has exceeded the maximum limit, then the next frame will receive 1300 bits along with the number of bits by which the Bit reservoir has exceeded the limit.

In the above-described operation 600 when the next frame is a small frame (small frames generally occur rarely), then more bits are allocated to the small frame from the Bit Reservoir. The number of extra bits that can be allocated to the small frame is dependent on two factors. One is the number of bits present in the Bit Reservoir and the other is the number of consecutive small blocks present in the input audio signal. Basically the strategy described in the above operation 600 is to remove bits from the long frames and to allocate the removed bits to the small frames as needed.

Figure 7 illustrates one example embodiment of operation 700 of reducing computational iterations during compression by a perceptual encoder to improve the operational efficiency of the perceptual audio coder. The operation 700 begins with step 710 by initializing common scalefac for the current frame. In some embodiments, the common scalefac is initialized using a common scalefac value of a previous frame adjacent to the current frame. In some embodiments, this is the common scalefac value obtained during the first call of the Rate Control Loop in the previous frame of the corresponding channel and is denoted as predicted common scalefac. In some embodiments, the initial value of the common scalefac is set to start common scalefac + 1 when the predicted common scalefac value is not greater than the common scalefac value. In some embodiments, the common scalefac includes a global gain for a given set of spectral values within the frame. The minimum value of common scalefac or the global gain is referred to as start common scalefac value. The value of quantizer change, which is the step-size for changing the value of common scalefac in the iterative algorithm, is set to 1.

At 720 counted bits associated with the current frame are computed. In some embodiments, computing counted bits includes qunatizing the spectrum of the current frame and then computing the number of bits required to encode the quantized spectrum of the current frame.

At 730 a difference between the computed counted bits and available bits are computed. In some embodiments, the available bits are the number of bits made available to encode the spectrum of the current frame. In some embodiments, the difference between the computed counted bits and the available bits are computed by comparing the computed counted bits with the available bits.

At 740 the computed difference is compared with a pre-determined MAXDIFF value. Generally, the value of pre-determined MAXDIFF is set to be in the range of about 300-500.

At 750 the common scalefac value and quantizer change value are reset based on the outcome of the comparison. In some embodiments, the common scalefac value is reset when the computed difference is greater than the pre-determined MAXDIFF, and the common scalefac value is changed based on the outcome of the comparison when the computed difference is less than or equal to the pre-determined MAXDIFF value.

In some embodiments, the changing of the common scalefac value based on the outcome of the comparison further includes storing the computed counted bits along with the associated common scalefac value, then comparing the counted bits with the available bits, and finally changing the common scalefac value based on the outcome of the comparison.

In some embodiments, changing the common scalefac value based on the outcome of the comparison further includes assigning a value to a quantizer change, and changing the common scalefac value using the assigned value to the quantizer change and repeating the above steps when the counted bits is greater than the available bits. Some embodiments include restoring the counted bits and outputting the common scalefac value when the counted bits is less than or equal to available bits.

In some embodiments, resetting the common scalefac value further includes computing predicted common scalefac value based on stored common scalefac value of the previous frame adjacent to the current frame, and resetting the common scalefac value. In case counted bits is greater than available bits, common scalefac is set to the start common scalefac value + 64, when the start common scalefac value + 64 is not greater than predicted common scalefac

value, otherwise common scalefac is set to predicted common scalefac and quantizer change is set to 64. Some embodiments include setting common scalefac to start common scalefac + 32, and further setting quantizer change to 32 when the counted bits is less than or equal to available bits and the common scalefac is not greater than start common scalefac + 32 and if predicted common scalefac is greater than the present common scalefac, recomputing counted bits. Further, some embodiments include setting the start common scalefac + 64 when the counted bits is less than or equal to available bits, and the common scalefac value is greater than the start common scalefac + 32 and if predicted common scalefac is greater than the present common scalefac, recomputing counted bits.

Figure 8 illustrates one example embodiment of operation 800 of stereo coding to improve sound quality according to the present invention. The operation 800 begins with step 810 by converting left and right audio signals into left and right digital audio signals, respectively. Step 820 divides each of the converted left and right digital audio signals into frames having a long-block frame length. In some embodiments, the long-block frame length includes 1024 samples of digital audio signal.

Step 830 includes partitioning each of the frames into corresponding multiple left and right short-blocks having short-block frame length. In some embodiments, the short-block frame-length includes samples in the range of about 100 to 300 samples of digital audio signal.

Step 840 includes computing left and right short-block characteristics for each of the partitioned left and right short-blocks. In some embodiments, the computing the short-block characteristics includes computing the sum and difference short-block characteristics by summing and subtracting respective samples of the digital audio signals in the left and right short-blocks. In some embodiments, computing the sum and difference short-block characteristics further includes computing sum and difference energies in each of the short-blocks in the left and right short-blocks by squaring each of the samples and adding the squared samples in each of the left and right short-blocks. In addition, the short-block energy ratio is computed for each of the short-blocks computed sum and difference energies, further determining a number of short-blocks

whose computed short-block energy ratio exceeds a pre-determined energy ratio value.

Step 850 includes encoding the stereo audio signal based on the computed short-block characteristics. In some embodiments, the encoding of the stereo signal includes using a sum and difference compression encoding technique to encode the left and right audio signals based on the determined number of short-blocks exceeding the pre-determined energy ratio value. In some embodiments, the pre-determined energy value is greater than 0.75 and less than 0.25.

Figure 9 shows an example of a suitable computing system environment 900 for implementing embodiments of the present invention, such as those shown in Figures 1-8. Various aspects of the present invention are implemented in software, which may be run in the environment shown in Figure 9 or any other suitable computing environment. The present invention is operable in a number of other general purpose or special purpose computing environments. Some computing environments are personal computers, server computers, hand held devices, laptop devices, multiprocessors, microprocessors, set top boxes, programmable consumer electronics, network PCS, minicomputers, mainframe computers, distributed computing environments, and the like. The present invention may be implemented in part or in whole as computer-executable instructions, such as program modules that are executed by a computer. Generally, program modules include routines, programs, objects, components, data structures and the like to perform particular tasks or implement particular abstract data types. In a distributed computing environment, program modules may be located in local or remote storage devices.

Figure 9 shows a general computing device in the form of a computer 910, which may include a processing unit 902, memory 904, removable storage 912, and non-volatile memory 908. Computer 910 may include – or have access to a computing environment that includes – a variety of computer-readable media, such as volatile 906 and non-volatile memory 908, removable and non-removable storages 912 and 914, respectively. Computer storage includes RAM, ROM, EPROM & EEPROM, flash memory or other memory technologies, CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic

cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium capable of storing computer-readable instructions. Computer 910 may include – or have access to a computing environment that includes – input 916, output 918, and a communication connection 920. The computer 910 may operate in a networked environment using a communication connection 920 to connect to one or more remote computers. The remote computer may include a personal computer, server, router, network PC, a peer device or other common network node, or the like. The communication connection 920 may include a local area network (LAN), a wide area network (WAN) or other networks.

Conclusion

The above-described invention increases compression efficiency by providing a technique to allocate bits between long and short-blocks. Also, the present invention significantly enhances the sound quality of the encoded audio signal by more accurately detecting an attack and reducing pre-echo artifacts caused by attacks. In addition, the present invention provides an audio coder that is computationally efficient and more accurate in switching between the normal and the M-S modes when the audio signal is a stereo signal.

The above description is intended to be illustrative, and not restrictive. Many other embodiments will be apparent to those skilled in the art. The scope of the invention should therefore be determined by the appended claims, along with the full scope of equivalents to which such claims are entitled.

WHAT IS CLAIMED IS:

1. An improved method for detecting an attack in an input audio signal to reduce a pre-echo artifact caused by an attack during compression encoding of the input audio signal, comprising:

converting the input audio signal into a digital audio signal;

dividing the digital audio signal into large frames having a long-block frame length;

partitioning each of the large frames into multiple short-blocks;

computing short-block audio signal characteristics for each of the short-blocks based on changes in the input audio signal;

comparing the computed short-block audio signal characteristics to a set of threshold values to detect a presence of the attack in each of the short-blocks; and

changing the long-block frame length of one or more large frames based on the outcome of the comparison to reduce the pre-echo artifact caused by the attack.

2. The method of claim 1, wherein detecting the attack comprises:

detecting a sudden increase in amplitude within the long-block frame length.

3. The method of claim 2, wherein the long-block frame length comprises 1024 samples of digital audio signal.

4. The method of claim 3, wherein the samples of digital audio signal comprise series of numbers.

5. The method of claim 3, wherein the long-block frame length comprises a frame length used when there is no attack in the input audio signal.

6. The method of claim 5, wherein the large frames comprise:

current and previous adjacent frames.

7. The method of claim 5, wherein the short-blocks comprise:

short-blocks having short-block frame lengths in the range of about 100 to 300 samples.

8. The method of claim 5, wherein computing the short-block audio signal characteristics further comprises:

computing inter-block differences; and

determining a maximum inter-block difference from the computed inter-block differences.

9. The method of claim 8, wherein computing the short-block audio signal characteristics further comprises:

computing inter-block ratios; and

determining a maximum inter-block ratio from the computed inter-block ratios.

10. The method of claim 9, wherein computing the inter-block differences comprises:

summing a square of differences between samples in adjacent short-blocks.

11. The method of claim 10, wherein computing the inter-block ratios comprises:

dividing the adjacent computed inter-block differences.

12. The method of claim 10, wherein comparing the computed short-block values to the set of threshold values comprises:

comparing the determined maximum inter-block difference and the maximum inter-block ratio to the set of threshold values.

13. The method of claim 10, wherein changing the long-block frame length comprises:

changing the long-block frame length of the one or more large frames including the attack includes changing the long-block frame length to multiple smaller frames having smaller frame lengths to restrict the attack to one or more smaller frames so that the pre-echo artifact caused by the attack does not spread to the adjacent larger frames.

14. The method of claim 13, wherein each of the smaller-frame lengths comprises about 100 to 300 samples of digital audio signal.

15. The method of claim 13, further comprising:

computing an average number of bits that can be allocated for each of the large frames;

computing a perceptual entropy for the current frame of audio samples;

computing a bit rate using a sampling frequency and the current frame length;

computing a reduction factor based on the computed bit rate and the perceptual entropy;

computing a reduced average number of bits that can be allocated for each of the large frames using the computed reduction factor;

computing remaining bits by subtracting the computed average number of bits with the computed reduced average number of bits;

if the current frame to be coded is a large frame, then allocating the reduced average number of bits to the current frame and storing the remaining bits in a Bit Reservior; and

if the current frame to be coded is a small frame, then allocating the reduced average number of bits along with the stored bits from the Bit Reservoir to the current frame.

16. The method of claim 15 further comprising:

repeating the above steps for a next adjacent frame.

17. The method of claim 16, wherein computing the average number of bits further comprises:

determining a bit rate of the input audio signal;

determining the long-block frame length of the large frame;

determining a sampling frequency of the input audio signal; and

computing the average number of bits that can be allocated for each of the large frames based on the determined bit rate, long-block frame length, and sampling frequency.

18. A method of reducing computation during quantization iterations for compression of an input audio signal to improve the efficiency of operation of a perceptual encoder, comprising:

initializing a commonscalefac value of a current frame;

initializing a quantizer change value of the current frame;

computing counted bits associated with the current frame;

computing a difference between the computed counted bits and available bits;

comparing the computed difference with the a pre-determined MAXDIFF value;

if the computed difference is greater than the pre-determined MAXDIFF value, then resetting the common scalefac value and quantizer change value; and

if the computed difference is less than or equal to the pre-determined MAXDIFF value, then changing common scalefac value based on the outcome of the comparison.

19. The method of claim 18, wherein the common scalefac value comprises:

a global gain for a given set of spectral values within a frame.

20. The method of claim 18, wherein a start common scalefac comprises a theoretical minimum value of the common scalefac.

21. The method of claim 18, wherein the quantizer change comprises a step size to arrive at a final value of common scalefac.

22. The method of claim 18, wherein initializing the common scalefac value comprises:

initializing the common scalefac value of the current frame with a predicted common scalefac.

23. The method of claim 18, wherein initializing the common scalefac comprises setting the value of common scalefac to start common scalefac + 1 when the predicted common scalefac is less than the start common scalefac.

24. The method of claim 18 wherein initializing the quantizer change comprises setting the value of quantizer change to 1.

25. The method of claim 18, wherein computing the counted bits associated with the current frame comprises:

quantizing spectrum of the current frame; and

computing number of bits required to encode the quantized spectrum of the current frame.

26. The method of claim 18, wherein available bits comprises:

number of bits made available to encode the spectrum of the current frame.

27. The method of claim 18, wherein the pre-determined MAXDIFF value is in the range of about 300 – 500.

28. The method of claim 18, wherein changing common scalefac value based on the outcome of the comparison further comprises:

storing the computed counted bits along with the associated common scalefac value;

comparing the counted bits with the available bits; and

changing the common scalefac value based on the outcome of the comparison.

29. The method of claim 18, wherein changing the common scalefac value based on the outcome of the comparison further comprises:

assigning a value to a quantizer change;

if the counted bits is greater than the available bits, then changing the common scalefac value using the assigned value to the quantizer change and repeating the above steps starting with the computing of the counted bits; and

if the counted bits is less than or equal to available bits, then restoring the counted bits and outputting the common scalefac value.

30. The method of claim 18, wherein resetting the common scalefac value and the quantizer change value further comprises:

computing predicted common scalefac value based on stored common scalefac value of the previous frame adjacent to the current frame;

if counted bits is greater than available bits and if the start common scalefac value + 64 is not greater than predicted common scalefac value, then resetting the common scalefac value to the start common scalefac value + 64;

if the counted bits is less than or equal to available bits and the common scalefac is not greater than start common scalefac + 32, then the common scalefac is set to start common scalefac+32, and the quantizer change is set to 32 and counted bits is recomputed if predicted common scalefac is greater than common scalefac; and

if the counted bits is less than or equal to available bits, and the common scalefac value is greater than the start common scalefac + 32, then the common scalefac is set to start common scalefac + 64 and counted bits is recomputed if predicted common scalefac is greater than common scalefac.

31. An improved method of compression encoding a stereo audio signal, including left and right audio signals, comprising:

converting the left and right audio signals into left and right digital audio signals, respectively;

dividing each of the left and right digital audio signals into frames having a long-block frame length;

partitioning each of the frames into corresponding multiple left and right short-blocks having short-block frame length;

computing left and right short-block characteristics for each of the partitioned left and right short-blocks; and

compression encoding the stereo audio signal based on the computed short-block characteristics.

32. The method of claim 31, wherein the long-block frame length comprises 1024 samples of digital audio signal.

33. The method of claim 32, wherein the samples of digital audio signal comprise series of numbers.

34. The method of claim 32, wherein the short-block frame length comprises: samples in the range of about 100 to 300 samples of digital audio signal.

35. The method of claim 34, wherein computing left and right short-block characteristics comprises:

computing sum and difference short-block characteristics by summing and subtracting respective samples of digital audio signals in the left and right short-blocks.

36. The method of claim 35, wherein computing the sum and difference short-block characteristics comprises:

computing sum and difference energies in each of the short-blocks in the left and right short-blocks by squaring each of the samples and adding the squared samples in each of the left and right short-blocks;

computing a short-block energy ratio using the respective short-block computed sum and difference energies;

determining a number of short-blocks whose computed short-block energy ratio exceeds a pre-determined energy ratio value; and

using a sum and difference compression encoding technique based on the determined number of short-blocks exceeding the pre-determined energy ratio value.

37. The method of claim 36, wherein the pre-determined energy ratio value is greater than 0.75 and less than 0.25.

38. A method for processing an audio signal, comprising:

converting the audio signal into a digital audio signal;

dividing the digital audio signal into large frames having a long-block frame length;

partitioning each of the large frames into multiple short-blocks;

computing short-block audio signal characteristics for each of the short-blocks based on changes in the input audio signal;

comparing the computed short-block audio signal characteristics to a set of threshold values to detect a presence of the attack in each of the short-blocks; and

changing the long-block frame length of one or more large frames based on the outcome of the comparison to reduce the pre-echo artifact caused by the attack.

39. The method of claim 38, wherein detecting the attack comprises:

detecting a sudden increase in amplitude within the long-block frame length.

40. The method of claim 38, wherein the long-block frame length comprises 1024 samples of digital audio signal.

41. The method of claim 40, wherein the samples of digital audio signal comprise series of numbers.

42. The method of claim 41, wherein the long-block frame length comprises a frame length used when there is no attack in the input audio signal.

43. The method of claim 41, wherein the short-blocks comprise:

short-blocks having short-block frame lengths in the range of about 100 to 300 samples.

44. The method of claim 41, wherein computing the short-block audio signal characteristics further comprises:

computing inter-block differences; and

determining a maximum inter-block difference from the computed inter-block differences.

45. An apparatus to detect an attack in an input digital audio signal to reduce a pre-echo artifact caused by the attack during compression encoding of the input digital audio signal, comprising:

a time frequency generator to receive the digital audio signal and divide the digital audio signal into large frames having a long-block frame length, and to further partition each of the large frames into multiple short-blocks; and

a transient detection module coupled to the time frequency generator to receive the multiple short-blocks and compute short-block audio signal characteristics for each of the received multiple short-blocks based on changes in the input digital audio signal, wherein the transient detection module compares the computed short-block audio signal characteristics to a set of threshold values to detect a presence of the attack in each of the multiple short-blocks, and the transient detection module further changes the long-block frame length of one or more large frames including the attack based on the outcome of the comparison, wherein the time frequency generator receives the changed one or more large frames and compresses the changed one or more large frames to reduce the pre-echo artifact caused by the attack.

46. The apparatus of claim 45, wherein the attack comprises:

a sudden increase in amplitude within the long-block frame length of the large frame of digital audio signal.

47. The apparatus of claim 46, wherein the long-block frame length comprises 1024 samples of digital audio signal.

48. The apparatus of claim 47, wherein the samples of digital audio signal comprise samples selected from the group consisting of series of numbers and bits.

49. The apparatus of claim 47, wherein long-block frame length comprises a frame length used when there is no attack in the input digital audio signal.

50. The apparatus of claim 47, wherein the large frames comprise; a current and a previous adjacent frame.

51. The apparatus of claim 50, wherein the short-blocks comprise a frame length in the range of about 100 to 300 samples.

52. The apparatus of claim 50, wherein the transient detection module further computes inter-block differences and determines a maximum inter-block difference from the computed inter-block differences.

53. The apparatus of claim 52, wherein the transient detection module further computes the inter-block differences by summing the samples in each of the short-blocks to obtain a short-block signal for each of the short-blocks, and further computes the inter-block differences by using the summed short-block signals of adjacent short-blocks.

54. The apparatus of claim 52, wherein the transient detection module further computes inter-block ratios and determines a maximum inter-block ratio from the computed inter-block ratios.

55. The apparatus of claim 54, wherein the transient detection module further computes inter-block ratios by dividing the adjacent computed inter-block differences.

56. The apparatus of claim 54, wherein the transient detection module compares the determined maximum inter-block difference and the maximum inter-block ratio to a set of threshold values to detect the presence of the attack.

57. The apparatus of claim 54, wherein the transient detection module changes the long-block frame length of the one or more large frames including the attack to multiple smaller frames having smaller frame lengths to restrict the attack to one or more smaller frames so that the attack does not spread to the adjacent large frames to reduce the pre-echo artifact caused by the attack.

58. The apparatus of claim 57, wherein each of the smaller frame lengths comprises samples in the range of about 100 to 200 samples of digital audio signal.

59. The apparatus of claim 54, further comprising:

a psychoacoustic model coupled to the transient detection module to compute a perceptual entropy for the current frame including samples of digital audio signal;

a quantizer coupled to the time frequency generator and the psychoacoustic model to receive the large and smaller frames including the samples of digital audio signal from the time frequency generator and the computed perceptual entropy from the psychoacoustic model, wherein the quantizer further comprises:

a Bit Allocator to compute an average number of bits that can be allocated to each of the received large frames, and to compute a bit rate and a reduction factor based on the computed bit rate, and the received perceptual entropy, the Bit Allocator further computing a reduced average number of bits that can be allocated for each of the large frames using the computed reduction factor, and further computing remaining bits by subtracting the computed average number of bits using the computed reduced average number of bits; and

a Bit Reservoir to receive the remaining bits, wherein the Bit Allocator allocates a reduced average number of bits to the current frame

and stores the remaining bits in the Bit Reservoir when the current frame is a large frame, and wherein the Bit Allocator further allocates the reduced number of bits along with the stored bits from the Bit Reservoir when the current frame is a small frame to improve the bit allocation between the large and small frames to enhance sound quality of the compressed audio signal.

60. The apparatus of claim 59, wherein the Bit Allocator repeats the bit allocation to a next adjacent frame.

61. The apparatus of claim 59, wherein the Bit Allocator determines the average number of bits by using the bit rate, the long-block frame length, and the sampling frequency.

62. The apparatus of claim 59, wherein the quantizer further comprises:

a memory to store a start common scalefac of the previous adjacent frame to use in computation of the current frame;

a Rate Control Loop to compute common scalefac of the current frame using the stored start common scalefac as a starting value during computation of iterations by the Rate control Loop to reduce the number of iterations required to compute the common scalefac of the current frame, and the Rate Control Loop further to compute counted bits using the common scalefac of the current frame; and

a comparator coupled to the Rate Control Loop to compare the computed count bits with available bits, wherein the Rate Control Loop changes computed common scalefac based on the outcome of the comparison.

63. The apparatus of claim 62, wherein the start common scalefac comprises:

a global gain for a given set of spectral values within the previous adjacent frame.

64. The apparatus of claim 63, wherein the count bits comprises:

bits required to encode a given set of spectral values for the current frame.

65. The apparatus of claim 62, wherein the Rate Control Loop intializes the common scalefac value with a predicted common scalefac obtained during a first call of the Rate Control Loop in the previous adjacent frame of a corresponding channel.

66. A computer readable medium having computer-executable instructions for an improved method for detecting an attack in an input audio signal to reduce a pre-echo artifact caused by an attack during compression encoding of the input audio signal, comprising:

converting the input audio signal into a digital audio signal;

dividing the digital audio signal into large frames having a long-block frame length;

partitioning each of the large frames into multiple short-blocks;

computing short-block audio signal characteristics for each of the short-blocks based on changes in the input audio signal;

comparing the computed short-block audio signal characteristics to a set of threshold values to detect a presence of the attack in each of the short-blocks; and

changing the long-block frame length of one or more large frames based on the outcome of the comparison to reduce the pre-echo artifact caused by the attack.

67. The computer readable medium as recited in claim 66, wherein detecting the attack comprises:

detecting a sudden increase in amplitude within the long-block frame length.

68. The computer readable medium of claim 67, wherein the long-block frame length comprises 1024 samples of digital audio signal.

69. The computer readable medium of claim 68, wherein the short-blocks comprise:

short-blocks having short-block frame lengths in the range of about 100 to 300 samples.

5

70. The computer readable medium of claim 67, wherein computing the short-block audio signal characteristics further comprises:

computing inter-block differences; and

determining a maximum inter-block difference from the computed inter-block differences.

71. The computer readable medium of claim 70, wherein computing the short-block audio signal characteristics further comprises:

computing inter-block ratios; and

determining a maximum inter-block ratio from the computed inter-block ratios.

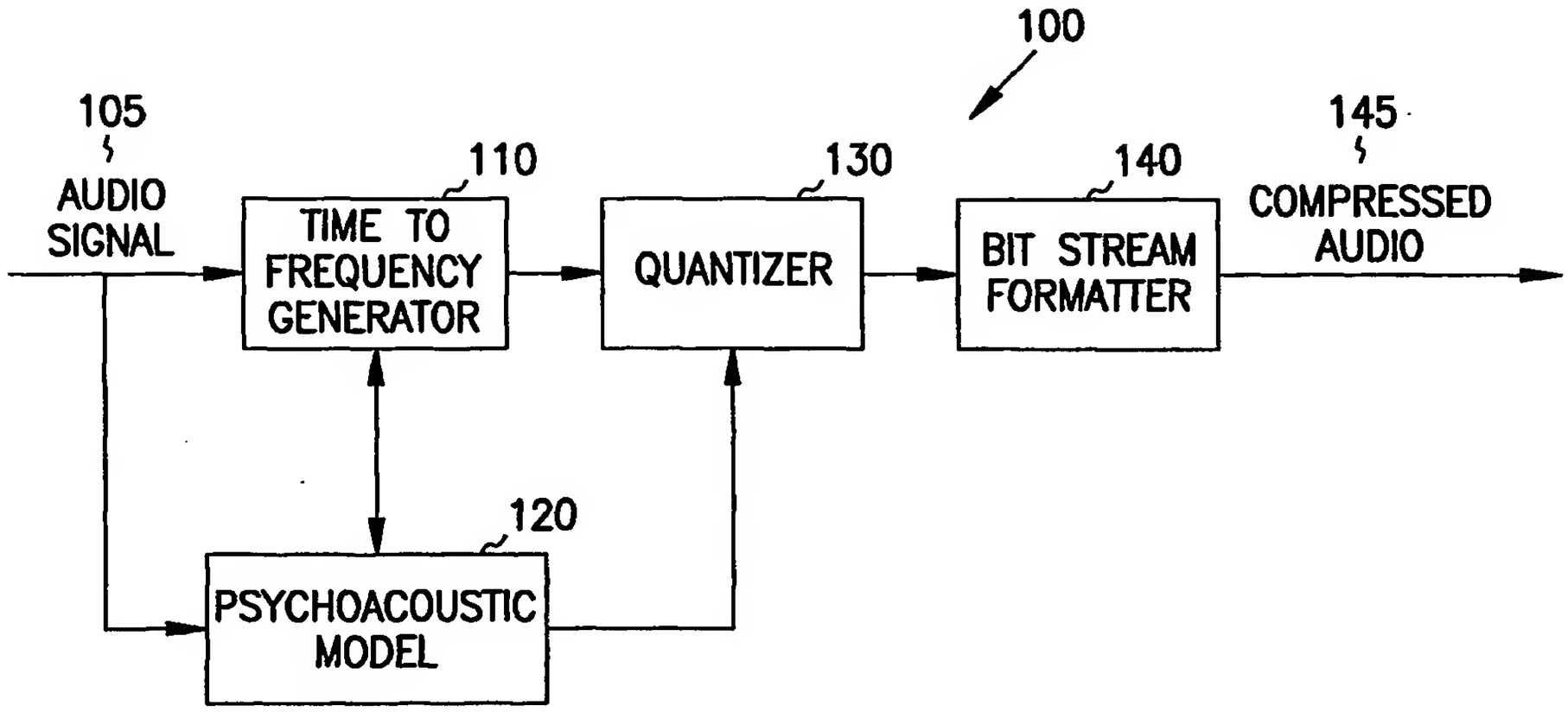


FIG. 1
(PRIOR ART)

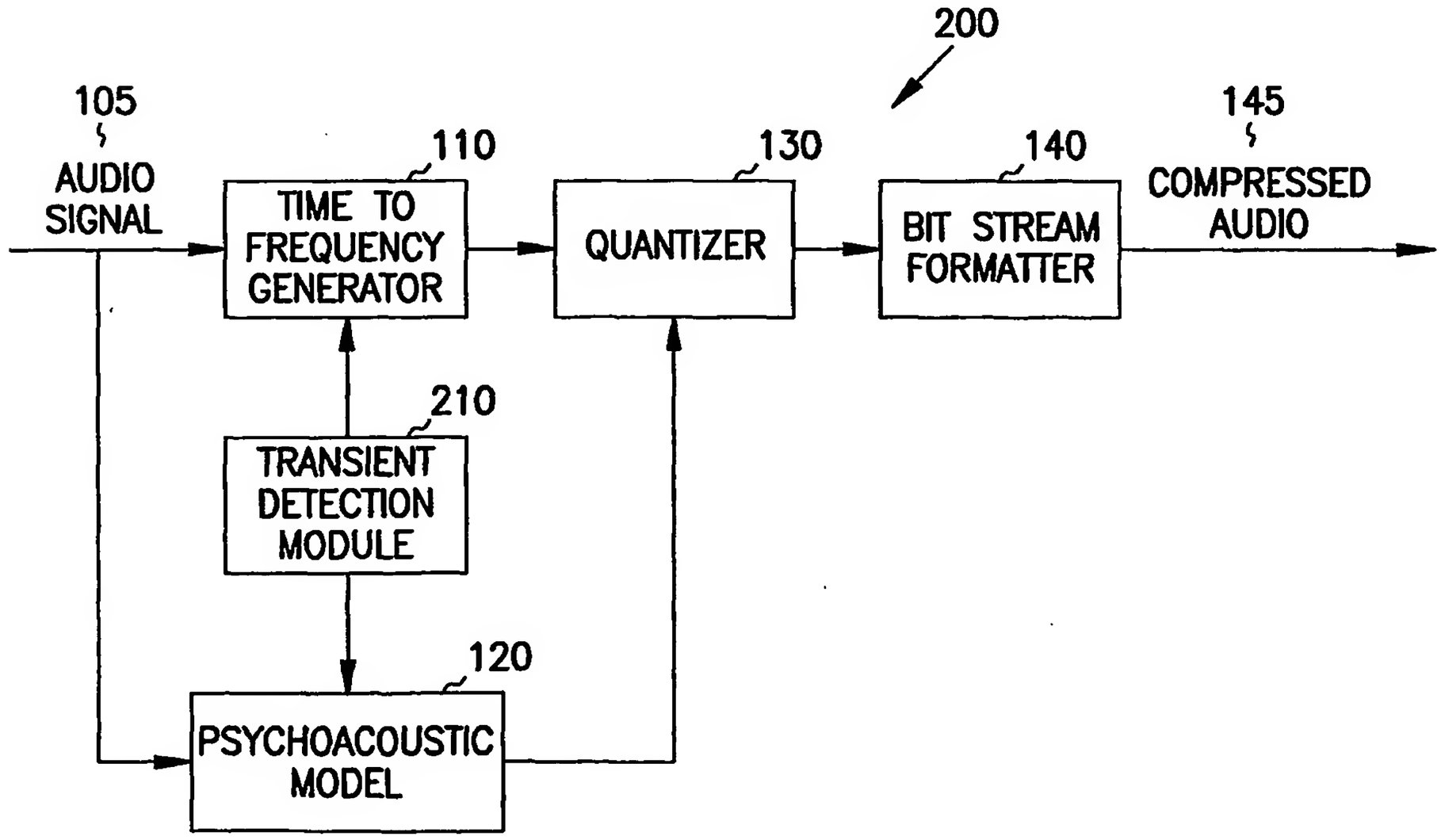


FIG. 2

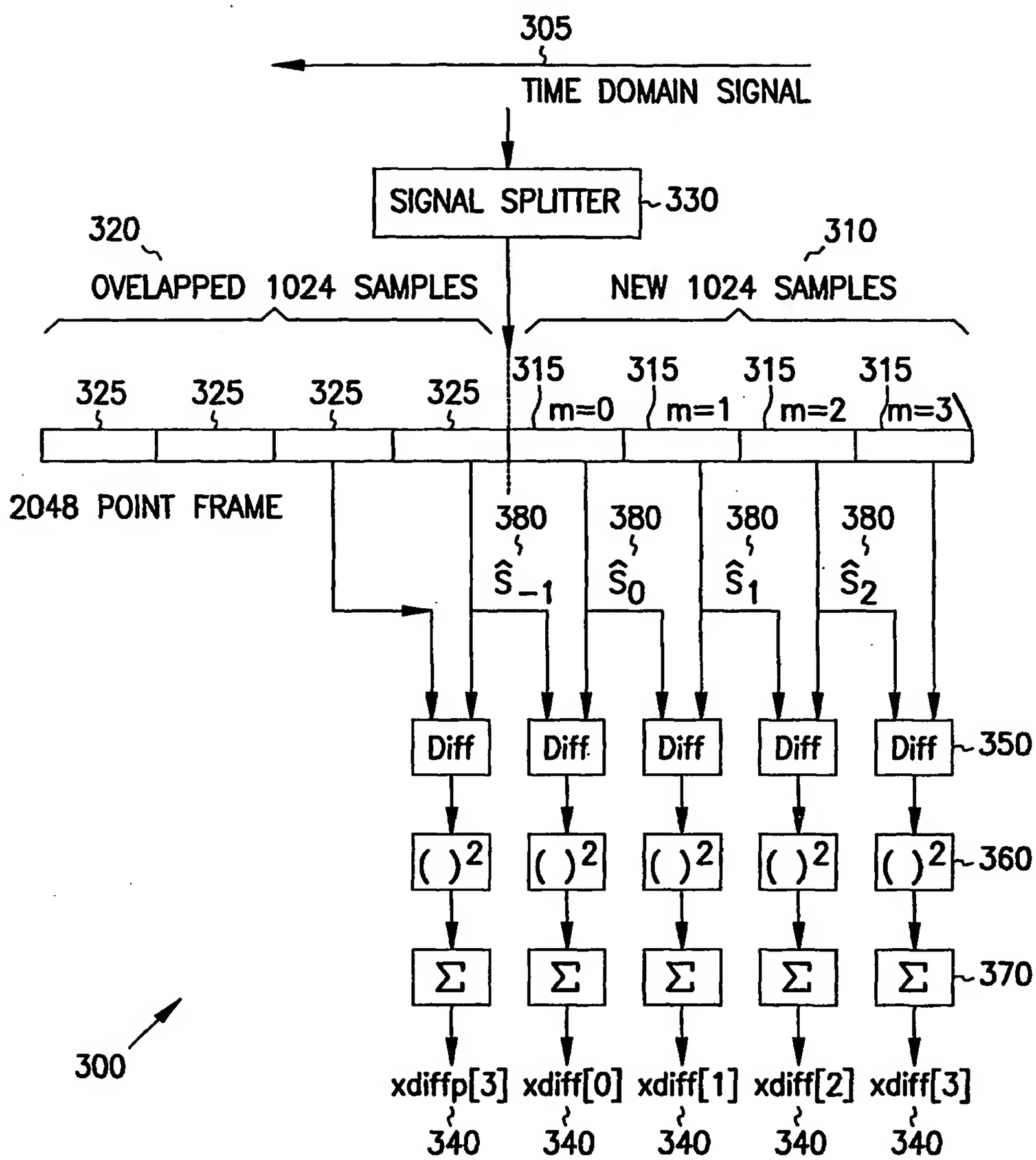

305
TIME DOMAIN SIGNAL
SIGNAL SPLITTER
330
320
OVELAPPED 1024 SAMPLES
310
NEW 1024 SAMPLES
315
315
315
315
325
325
325
325
m=0
m=1
m=2
m=3
2048 POINT FRAME
380
380
380
380
$\hat{S}_{-1}$
$\hat{S}_0$
$\hat{S}_1$
$\hat{S}_2$
Diff
Diff
Diff
Diff
Diff
350
$(\ )^2$
360
Σ
370
300
xdiffp[3]
xdiff[0]
xdiff[1]
xdiff[2]
xdiff[3]
340
340
340
340
340

FIG. 3

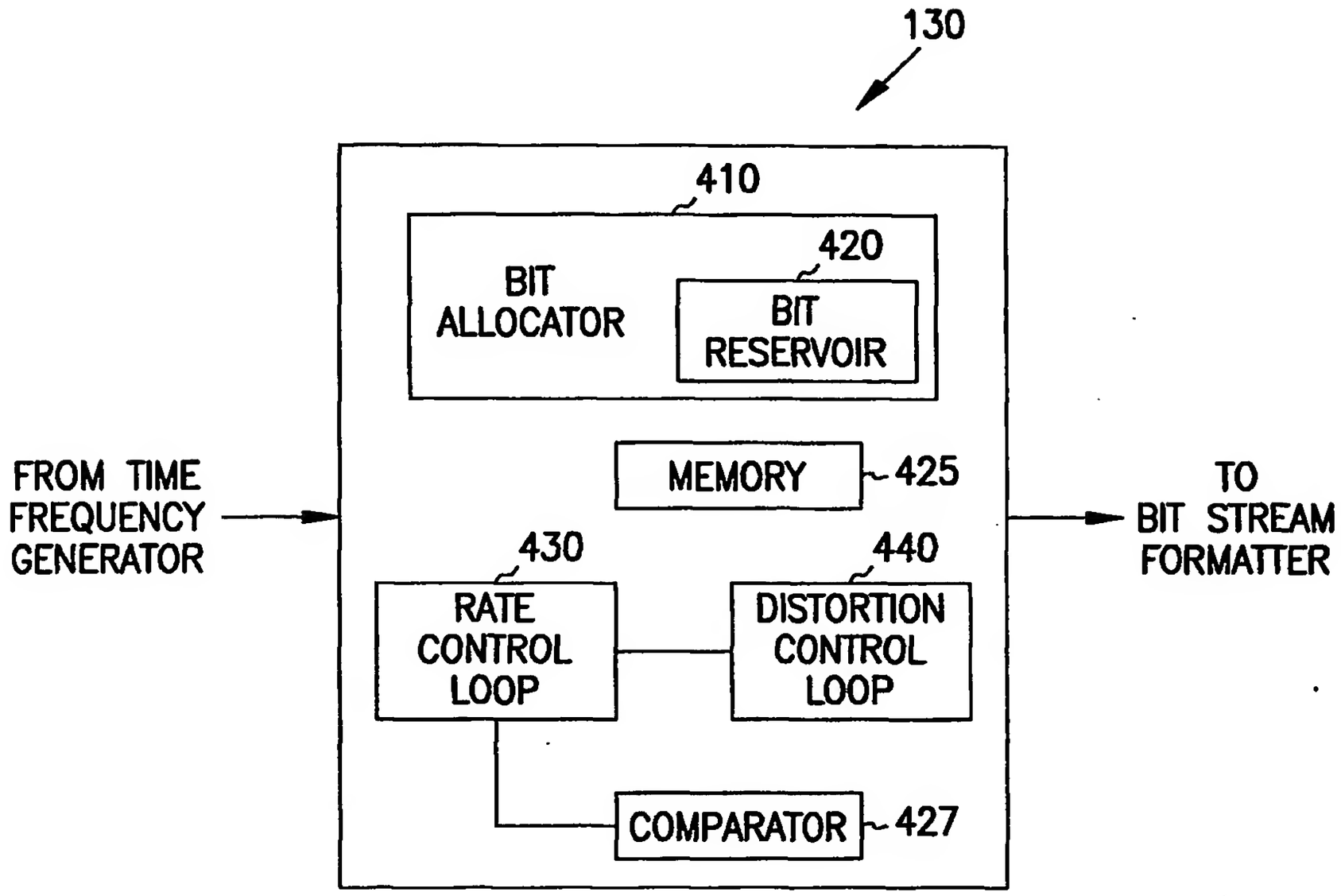

130
410
420
BIT ALLOCATOR
BIT RESERVOIR
MEMORY
425
FROM TIME FREQUENCY GENERATOR
TO BIT STREAM FORMATTER
430
440
RATE CONTROL LOOP
DISTORTION CONTROL LOOP
COMPARATOR
427

FIG. 4

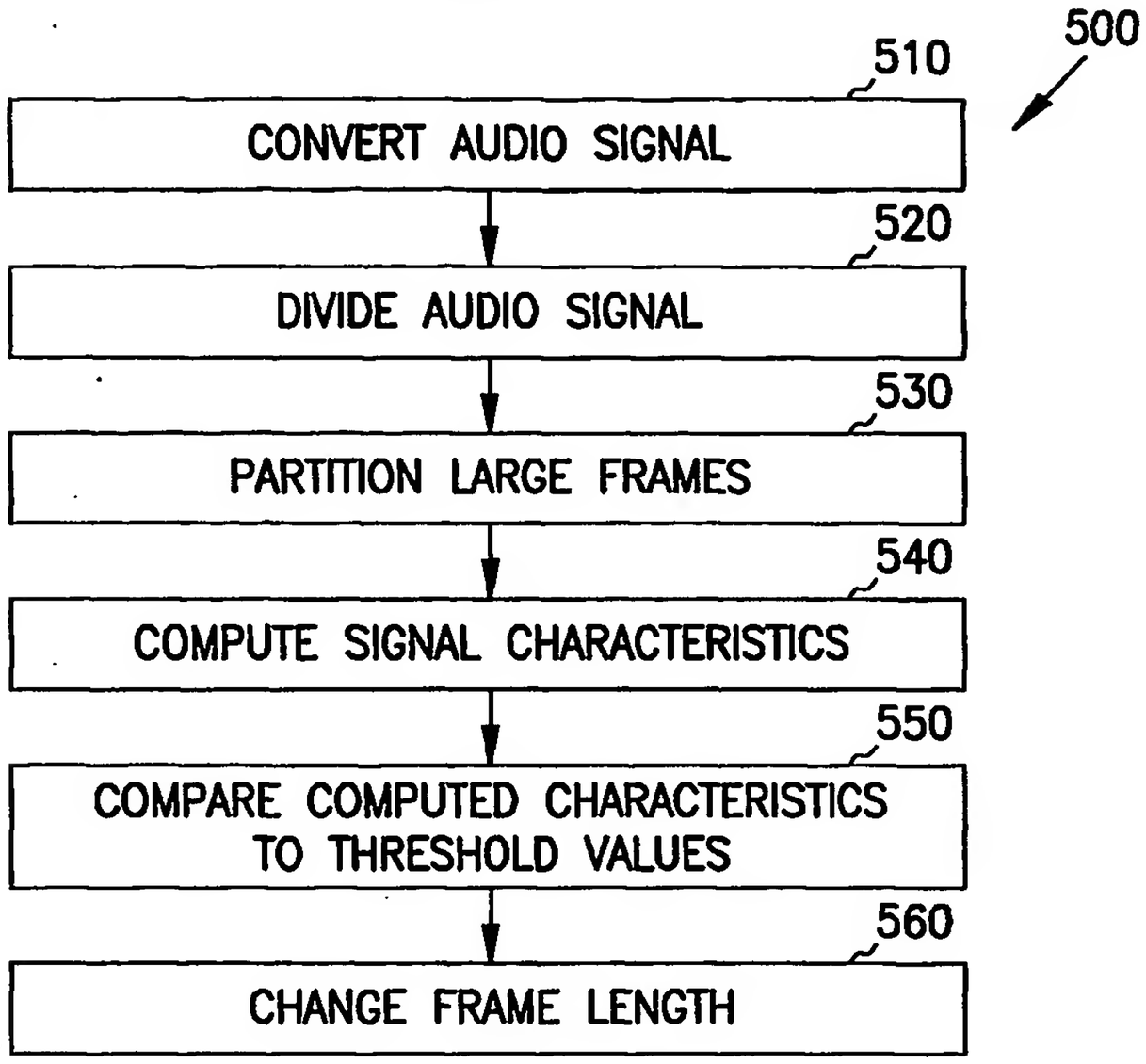

500
510
CONVERT AUDIO SIGNAL
520
DIVIDE AUDIO SIGNAL
530
PARTITION LARGE FRAMES
540
COMPUTE SIGNAL CHARACTERISTICS
550
COMPARE COMPUTED CHARACTERISTICS TO THRESHOLD VALUES
560
CHANGE FRAME LENGTH

FIG. 5

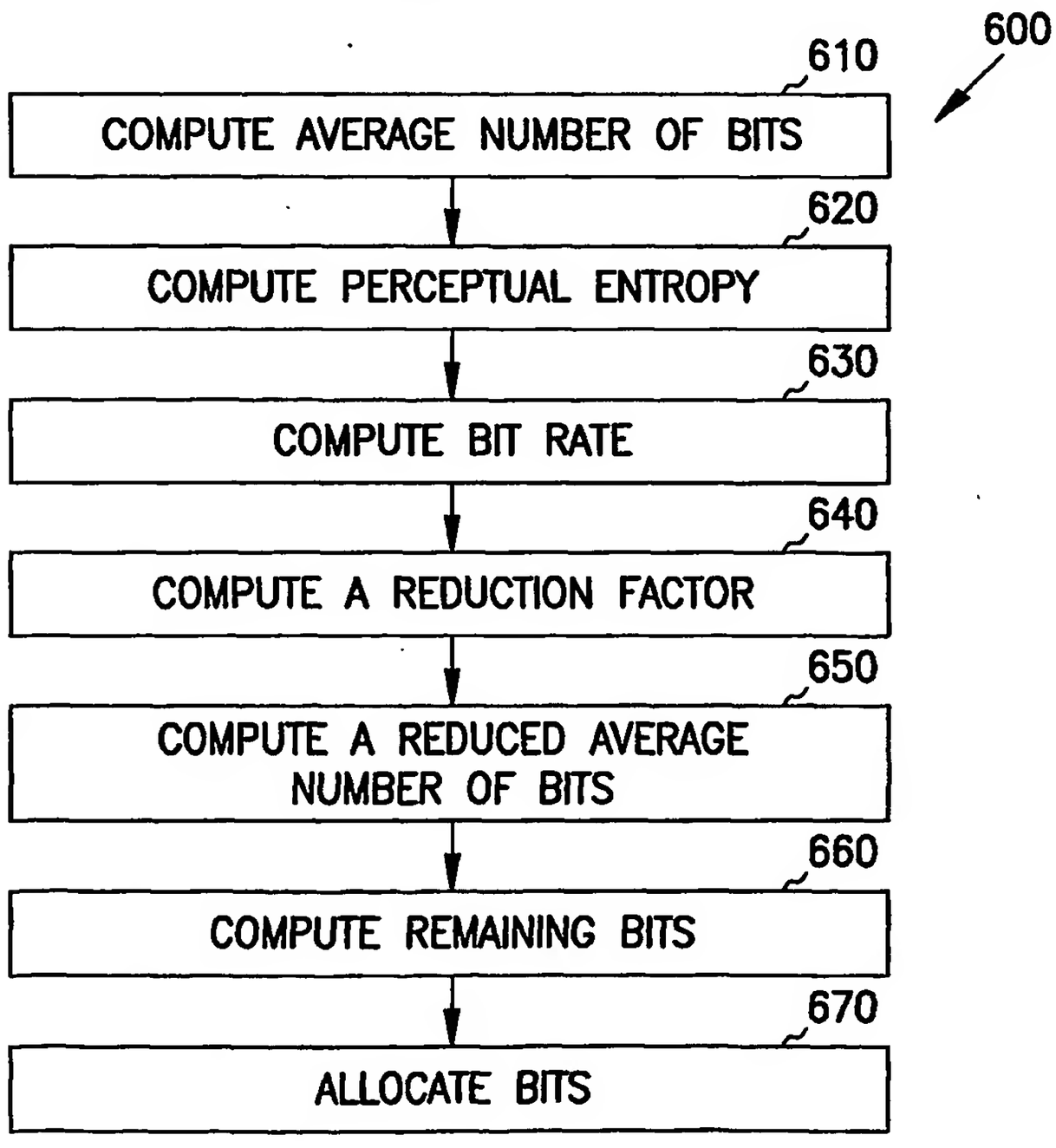

600
610
COMPUTE AVERAGE NUMBER OF BITS
620
COMPUTE PERCEPTUAL ENTROPY
630
COMPUTE BIT RATE
640
COMPUTE A REDUCTION FACTOR
650
COMPUTE A REDUCED AVERAGE NUMBER OF BITS
660
COMPUTE REMAINING BITS
670
ALLOCATE BITS

FIG. 6

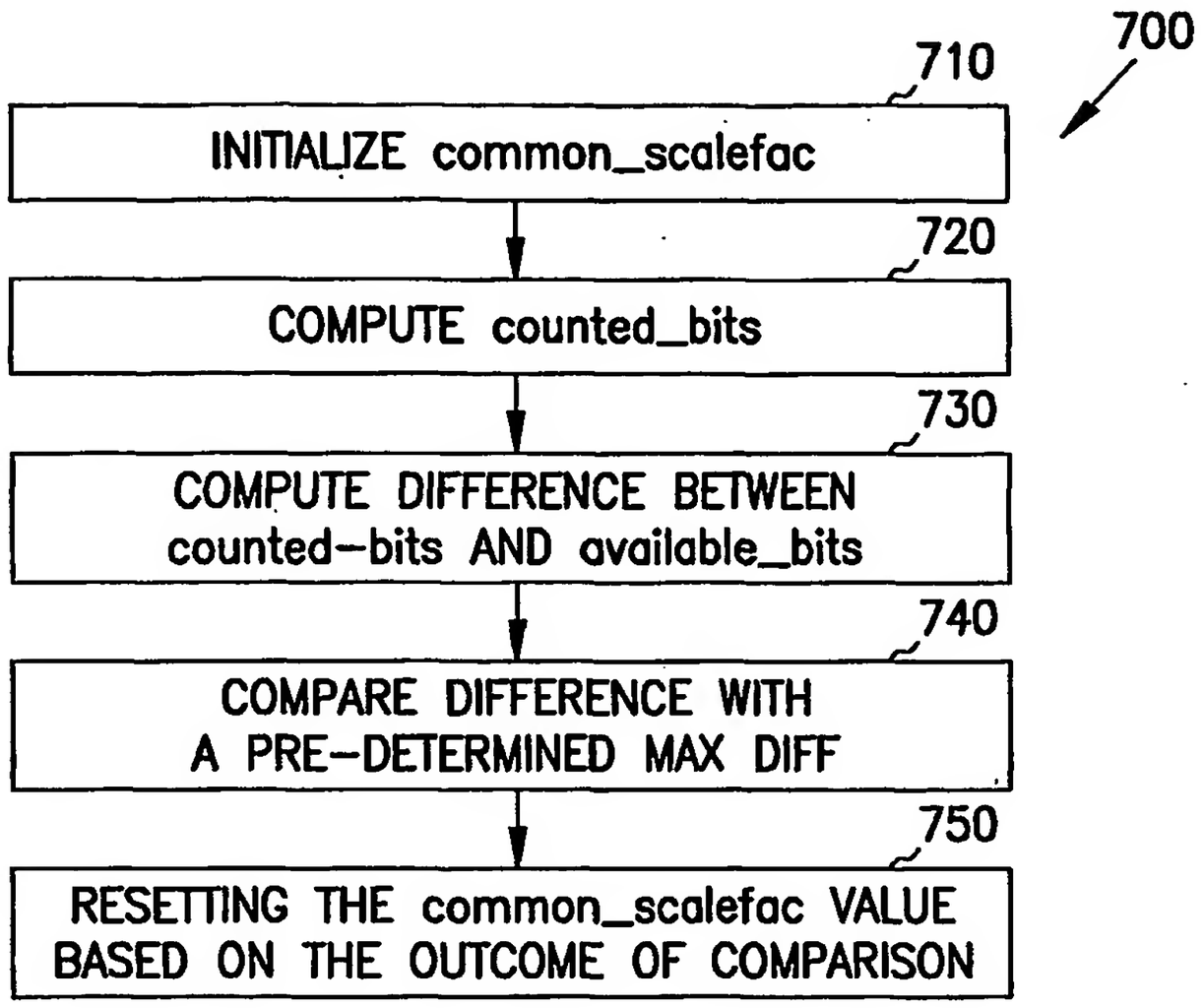

700
710
INITIALIZE common_scalefac
720
COMPUTE counted_bits
730
COMPUTE DIFFERENCE BETWEEN
counted–bits AND available_bits
740
COMPARE DIFFERENCE WITH
A PRE–DETERMINED MAX DIFF
750
RESETTING THE common_scalefac VALUE
BASED ON THE OUTCOME OF COMPARISON

FIG. 7

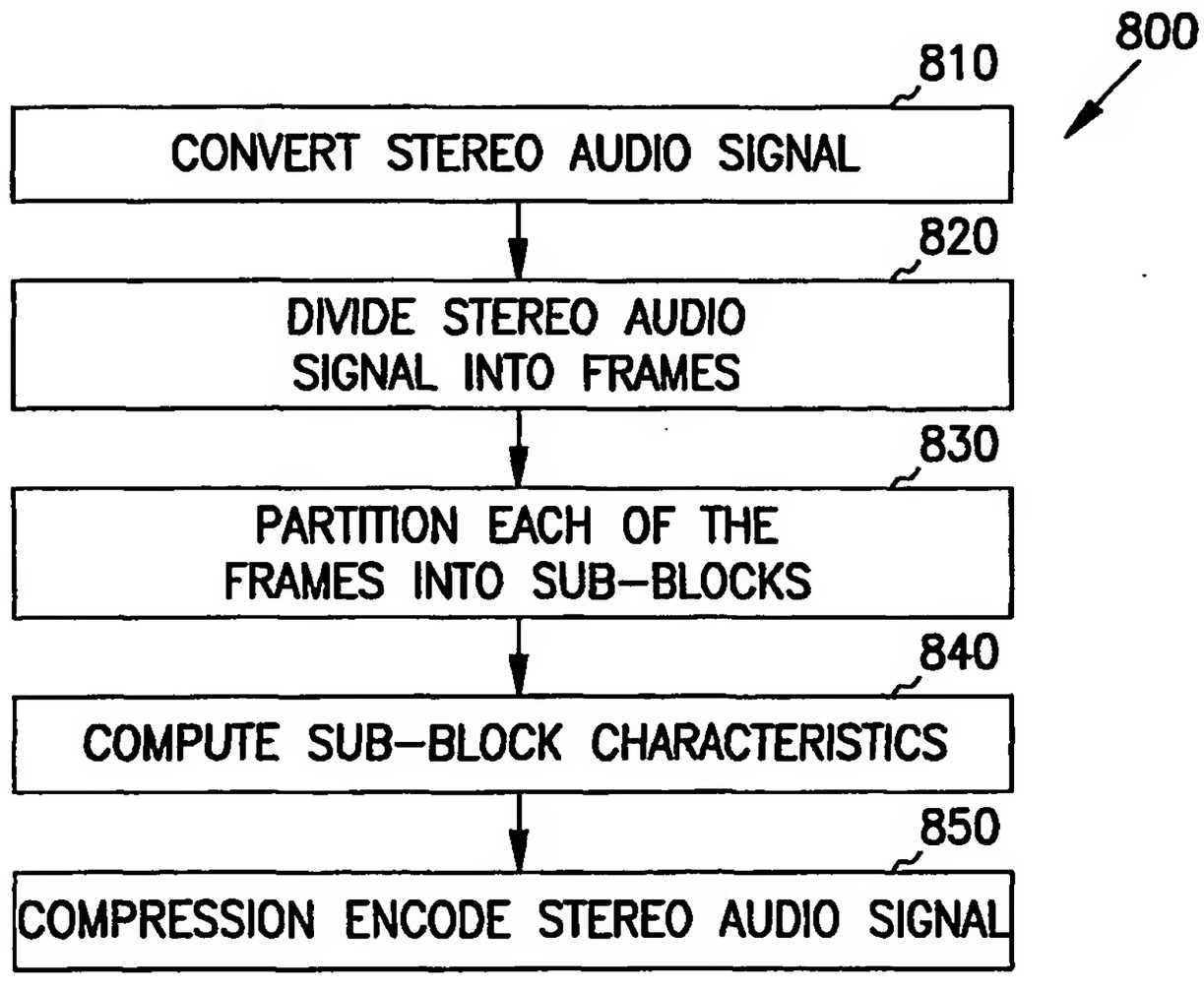

800
810
CONVERT STEREO AUDIO SIGNAL
820
DIVIDE STEREO AUDIO
SIGNAL INTO FRAMES
830
PARTITION EACH OF THE
FRAMES INTO SUB–BLOCKS
840
COMPUTE SUB–BLOCK CHARACTERISTICS
850
COMPRESSION ENCODE STEREO AUDIO SIGNAL

FIG. 8

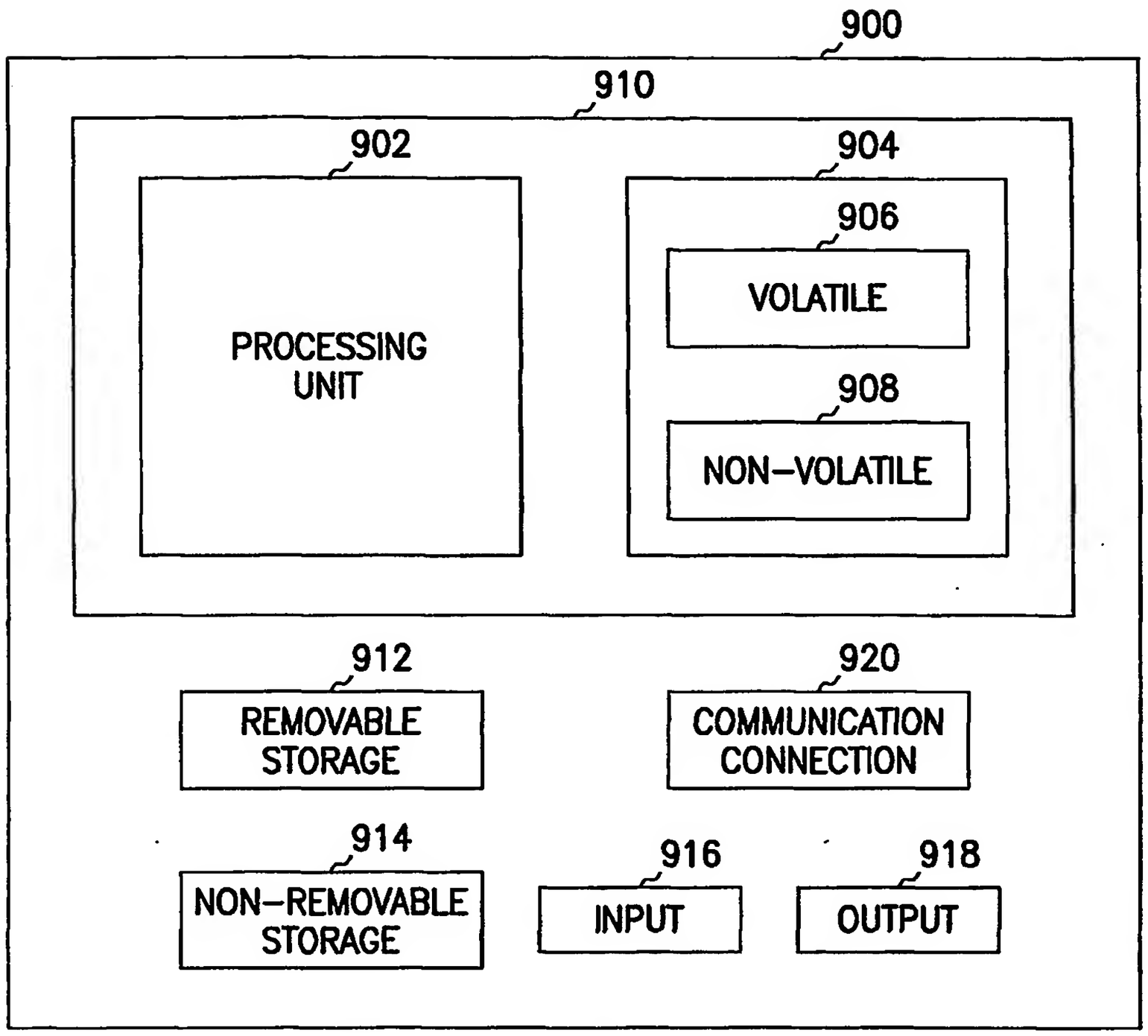

900
910
902
PROCESSING UNIT
904
906
VOLATILE
908
NON–VOLATILE
912
REMOVABLE STORAGE
920
COMMUNICATION CONNECTION
914
NON–REMOVABLE STORAGE
916
INPUT
918
OUTPUT

FIG. 9

INTERNATIONAL SEARCH REPORT

Inte Application No
PCT/IB 01/01371

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G10L19/02 G10L19/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G10L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 725 493 A (AT & T CORP) 7 August 1996 (1996-08-07) abstract column 1, line 12-39 column 2, line 11-36 column 3, line 12-23,33-42 column 3, line 53 -column 4, line 6 ---	1-7, 38-43, 45-51, 66-69
X	US 5 848 391 A (BOSI ET AL) 8 December 1998 (1998-12-08) abstract; figures 1,8A-B column 2, line 8-20,40-46 column 4, line 4-32 --- -/--	1-5, 38-42, 45-49, 66-68

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
14 March 2002	03. 07. 2002

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Quélavoine, R

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Inter ·nal Application No
PCT/IB 01/01371

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 612 158 A (MATSUSHITA ELECTRIC IND CO LTD) 24 August 1994 (1994-08-24)	1,2,38, 39,45, 46,66,67
A	abstract column 1, line 57 -column 2, line 5 column 2, line 16-32 column 2, line 52 -column 3, line 20	8-13,44, 52-57, 70,71
X	US 5 819 214 A (SUZUKI ET AL) 6 October 1998 (1998-10-06) abstract; figures 4,5A-C column 2, line 12-56 column 2, line 66 -column 3, line 37 column 8, line 53 -column 9, line 8 column 9, line 29-64	1,2,38, 39,45, 46,66,67
X	US 5 825 320 A (MIYAMORI ET AL) 20 October 1998 (1998-10-20) abstract; figures 2A-B column 4, line 6 -column 5, line 48	1,2,38, 39,45, 46,66,67

INTERNATIONAL SEARCH REPORT

ɔnal application No.
PCT/IB 01/01371

Box I Observations where certain claims were found unsearchable (Continuation of item 1 of first sheet)

This International Search Report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the International Application that do not comply with the prescribed requirements to such an extent that no meaningful International Search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. ☐ As all required additional search fees were timely paid by the applicant, this International Search Report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this International Search Report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this International Search Report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

1-17, 38-71

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest.

☐ No protest accompanied the payment of additional search fees.

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. Claims: 1-17,38-71

detecting an attack in long blocks of input audio signal, changing to shorter blocks if needed, to avoid a pre-echo artifact.

2. Claims: 18-30

Bit allocation algorithm between long and short blocks for audio compression with a bit reservoir.

3. Claims: 31-37

sum and difference compression encoding on a stereo audio signal

INTERNATIONAL SEARCH REPORT

Information on patent family members

Inten Application No
PCT/IB 01/01371

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0725493 A	07-08-1996	US 5701389 A	23-12-1997
		EP 0725493 A2	07-08-1996
		JP 3276835 B2	22-04-2002
		JP 8256062 A	01-10-1996
		TW 441197 B	16-06-2001
US 5848391 A	08-12-1998	AU 710868 B2	30-09-1999
		AU 7696196 A	09-02-1998
		CA 2260033 A1	22-01-1998
		WO 9802971 A1	22-01-1998
		EP 0910900 A1	28-04-1999
		JP 3171598 B2	28-05-2001
EP 0612158 A	24-08-1994	JP 3088580 B2	18-09-2000
		JP 6242797 A	02-09-1994
		DE 69423803 D1	11-05-2000
		DE 69423803 T2	03-08-2000
		EP 0612158 A1	24-08-1994
		US 5651089 A	22-07-1997
US 5819214 A	06-10-1998	JP 3186307 B2	11-07-2001
		JP 7038443 A	07-02-1995
		DE 69429687 D1	14-03-2002
		EP 0615348 A1	14-09-1994
		US 6046190 A	04-04-2000
US 5825320 A	20-10-1998	JP 9261063 A	03-10-1997